

**MA4830 Realtime Software for Mechatronic Systems**

**Major CA Project**

**Submitted To: Assoc. Prof Seet Gim Lee, Gerald**

**Group Members:**

Left to Right:

**Ng Ee Meng (U1420377G)**

**Khoo Kar Keng (U1420494C)**

**Joyce Lim Xin Yan (U1521972L)**

**Tan Jin Cai (U1421759E)**

**Joseph Ng Rong En (U1422346L)**

**Phillip Puan Jun Kit (U1420409F)**

Table of Contents

[1. Introduction 2](#_Toc498634612)

[2. Description of Program 2](#_Toc498634613)

[3. Limitations 2](#_Toc498634614)

[4. User Manual 3](#_Toc498634615)

[4.1 Execute Program 3](#_Toc498634616)

[4.2 General Help Menu 3](#_Toc498634617)

[4.3 Keyboard Input 5](#_Toc498634618)

List of Figures

Figure 1: Compile and Execute Code 2

Figure 2: Program Help Menu 2

Figure 3: Showing current DAC0 configurations 3

Figure 4: Showing current ADCs and their statuses 3

Figure 5: Menu for keyboard commands 4

Figure 6: Changing type of waveform using keyboard 4

Figure 7: Changing frequency of waveform using keyboard 5

Figure 8: Changing amplitude of waveform using keyboard 5

Figure 9: Changing the amplitude value of waveform using keyboard 6

Figure 10: Turning DAC[0] OFF/ON using keyboard 6

Figure 11: Resetting waveform to default values using keyboard 7

Figure 12: Export Configuration into file.txt 7

Figure 13: Importing saved waveform configuration 8

Figure 14: Termination of program with CTRL-C 8

Figure 15: Original Triangular Waveform (example) 9

Figure 16: Changed Frequency of Original 9

Figure 17: Changed Frequency & Amplitude of Original 9

Figure 18: Changed Frequency, Amplitude & Mean of Original 9

Figure 19: Sine Wave Output 10

Figure 20: Square Wave Output 10

# Introduction

QNX is a very useful OS for doing real-time programming. It is used for various time-critical tasks, which are often present in a Mechatronic system. However, without the proper peripherals, its functionality is limited to the computing space.

Throughout the course of MA4830 Realtime Software for Mechatronic Systems, the computer is connected to a PCI-DAS1602/16 board. The PCI-DAS1602/16 is a multifunction measurement and control board designed to operate in computers with PCI bus accessory slots.

The use of the various peripherals, such as General Purpose Input/Output (GPIO), Analog to Digital (ADC) and Digital to Analog (DAC) converters on the PCI-DAS1602/16 board allows us to interface QNX with the external environment. The robustness of QNX and the rich functionalities of the PCI-DAS1602/16 board allows a myriad of tasks to be undertaken.

To demonstrate both the capabilities of the OS and the PCI board, we have programmed a application for wave generation. This application takes in input from both a user interface and a set of physical peripherals, such as potentiometers and flip switches. Using these inputs, a waveform of choice is being produced by the system.

# Description of Program

The program interfaces with the PCI-DAS1602 board to produce waveform (either sinusoidal, triangular or square waveform) from the board’s DAC.

The main parameters of DAC are waveform type, frequency, amplitude, mean and isOn settings. The range for the inputs are:

1. Frequency – (ADC) from more than 0.00122Hz to less than 1750Hz, (1.22E-3, 1750) Hz.

(Keyboard) from more than 0 to less than 1750Hz, (0, 1750) Hz.

1. Mean - (ADC) from more than -9.8V to less than 9.8V, (-9.8, 9.8) V

(Keyboard) from more than -10V to less than 10V, (-10, 10) V

1. Amplitude - (ADC) from more than or equal to 0V to less than 9.8V, [0, 9.8) V

(Keyboard) from more than or equal to 0V to less than 10V, [0, 10) V

The user can use command line arguments to initialise starting parameters. However, the following order must be met:

<program\_name.exe> <waveform type> <frequency> <mean> <amplitude> <isOn>

Waveform type argument can be “-sin” (sine wave), “-tri” (triangular wave) or “squ” (square wave). Frequency, mean and amplitude are floating point number and as such floating-point values can be inputted. The last argument is isOn argument where user insert 0/1 to turn off/on the DAC upon entry to the program.

Example execution is:

./wavegen -sin 1000.3 2.4 5 1

To change the parameters, the program takes in user input from the keyboard and/or from the potentiometers (using the board’s ADCs) and switches connected to the PCI board.

For the digital switches, bit description is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Bit number | Bit 3 | Bit 2 | Bit 1 and 0 |
| Description | Turn on/off parameters’ input acquisition from ADC and digital switches | Select the parameter to be changed by potentiometer 0 (ADC0) | Select the waveform type and change DAC operation |
| Bit description | 0 – Input acquisition from keyboard  1 – Input acquisition from ADC and digital switches (any keyboard input will be overwritten by continuous ADC acquisition) | 0 – Changes mean  1 – Changes amplitude | 0x00 – DAC is off  0x01 – Sine wave  0x02 – Triangular wave  0x03 – Square wave |

The formula used for ADC (16 bit) conversion to frequency, amplitude or mean values are:

ADC0

ADC1 (Dedicated to frequency)

HIGHESTFREQ is 1750Hz.

The user can also export and import the DAC configuration to text file in the format of command line statement. Further descriptions are in the user manual.

The novelties of the program are:

# Limitations

# User Manual

The user manual constitutes of the actual inputs using the keyboard, and the outputs on the screen and on the oscilloscope.

## 4.1 Execute Program

Figure 1: Compile and Execute Code

To compile the code and execute it, type in the commands as shown in Figure 1. In the execution, flags can be added to select type of wave form (-sin: sine, -tri: triangular, -squ: square), frequency, mean, amplitude and turning on/off DAC[0] (0: OFF, 1: ON) respectively. If no flags are added, the default settings will be sine wave, with frequency = 1 Hz, mean = 0 V, amplitude = 1 V and DAC0 is off.

## 4.2 General Help Menu

Figure 2: Program Help Menu

In Figure 2, it depicts the help menu when the program has been executed. The commands to be entered ranges from numerical values of 1-7. If input command is out of this range, “Invalid Choice” will be printed.

The following figures, Figure 3 and 4, illustrate how to get the various statuses and configurations of the DAC and ADCs.

Figure 3: Showing current DAC0 configurations

|  |  |
| --- | --- |
| **Keyboard Input:** 1  **Switches Input:** By turning the potentiometer, the frequency, amplitude or mean can be adjusted. | **Output:** Configurations of DAC0 which includes waveform types, number of samples per period, output resolution (V) and **current** values of frequency (Hz), amplitude (V) and mean (V). |



Figure 4: Showing current ADCs and their statuses

|  |  |
| --- | --- |
| **Keyboard Input:** 2 | **Output:** Shows the current ADC values and the DAC0 configurations. |

In addition, Figure 4 depicts how to control the switches to achieve desired configurations.

## 4.3 Keyboard Input

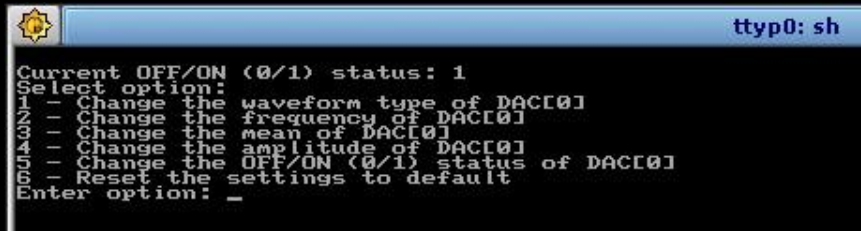
In Figures 5-11, they show how to change several values by using keyboard input. For these to occur, the PCI board must be switched off or the values of the board will overwrite those from the keyboard. From Figure 1, enter command **“3”** in the help menu to allow the program to change configurations by using the keyboard, and the screen is shown in Figure 5.

Figure 5: Menu for keyboard commands

The menu for keyboard commands are shown in Figure 5. Numerical commands ranging from 1-6 can be keyed in to produce various functions, which will be explained in Figures 6-11.

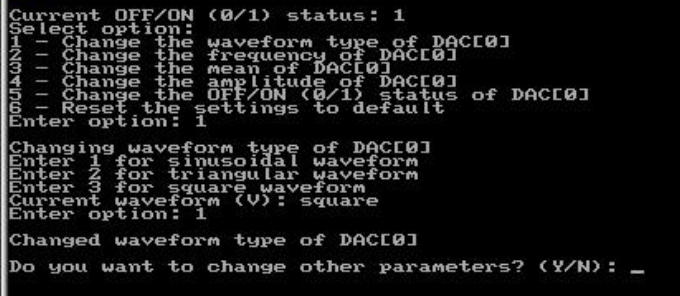


Figure 6: Changing type of waveform using keyboard

Upon entering command **“1”** from the menu for keyboard commands, the waveform type of DAC[0] can be changed. Figure 6 illustrates the current waveform type and commands to change the waveform to desired shapes. If the number entered is not an integer ranging from 1-3, an error message “Invalid input for waveform. (Must be integer)” will be printed.

|  |  |
| --- | --- |
| **Keyboard Input:** 1 | **Output:** Changes current waveform to sine wave |

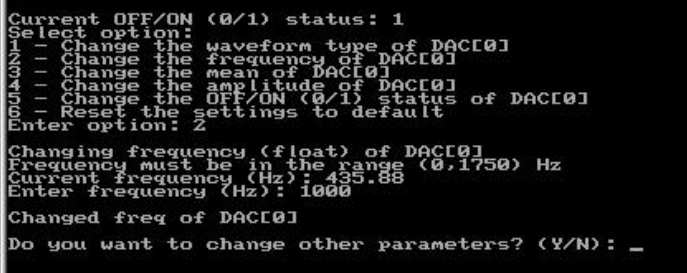
To change the frequency of the waveform of DAC[0] by using keyboard, input command **“2”** in the keyboard menu. Then, any value within the range can be used, as seen from Figure 7. If the value is out of range, an error message “Frequency is not changed” will be printed.

Figure 7: Changing frequency of waveform using keyboard

|  |  |
| --- | --- |
| **Keyboard Input:** 1000 | **Output:** Changes current frequency to 1000 Hz. |

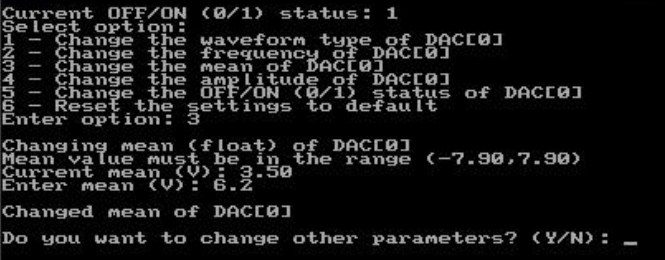
Similarly, in order to change the mean value of the waveform, command **“3”** is to be keyed into the keyboard menu, as shown in Figure 8. The mean value can then be changed. If the mean value entered is not within range, an error message “Mean value is not changed” will be seen.

Figure 8: Changing amplitude of waveform using keyboard

|  |  |
| --- | --- |
| **Keyboard Input:** 6.2 | **Output:** Changes current mean value to 6.2 V. |

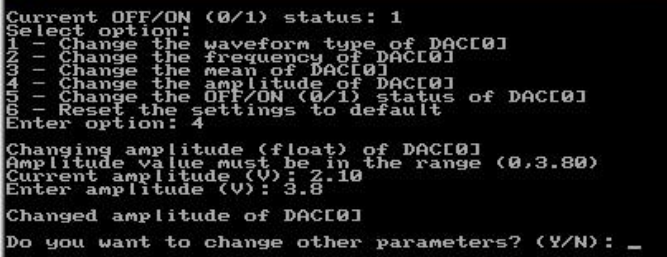
Also, to change the amplitude value, enter command **“4”** in the keyboard menu. Figure 9 demonstrates how to change the amplitude value of the waveform. If an error message of “Amplitude value is not changed” is seen, it means that the new amplitude value if out of range.

Figure 9: Changing the amplitude value of waveform using keyboard

|  |  |
| --- | --- |
| **Keyboard Input:** 3.8 | **Output:** Changes current amplitude value to 3.8 V. |

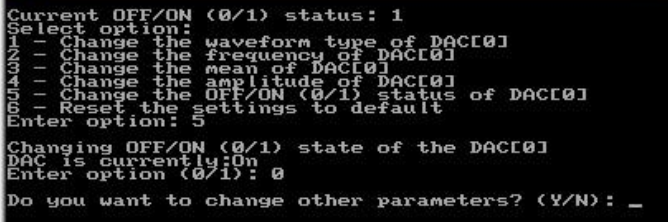
Turning off DAC[0] can be achieved by entering command **“5”** in the keyboard menu. The current state of the DAC is shown in Figure 10. If the new option entered is not 0 or 1, an error message “Invalid options for OFF/ON” will be printed. Also, if the current state and new input state are the same, the program will feedback to the user that the DAC is already off or on.

Figure 10: Turning DAC[0] OFF/ON using keyboard

|  |  |
| --- | --- |
| **Keyboard Input:** 0 | **Output:** If DAC[0] is ON, turn DAC[0] OFF. If DAC[0] is OFF, notify user that DAC[0] is already OFF. |

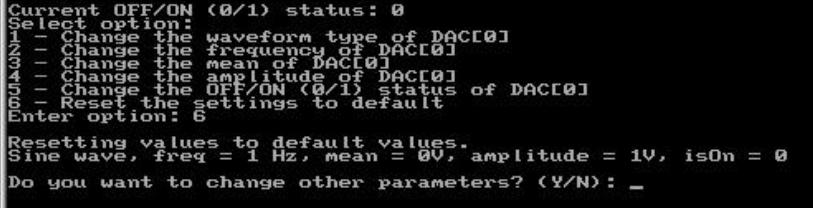
If the user desires to reset the waveform to default values, this can be done through the keyboard menu, as depicted in Figure 11.

Figure 11: Resetting waveform to default values using keyboard

|  |  |
| --- | --- |
| **Keyboard Input:** 6 | **Output:** Changes all previous values into default settings of sine wave, frequency = 1 Hz, mean = 0 V, amplitude = 1 V, and DAC to OFF. |

## 4.4 Export and Import Data Configurations

The program is capable of exporting and importing saved waveform configurations into a file, however, the PCI board must be turned off during importing of data importation to ensure that the current values on the board do not overwrite the saved file.

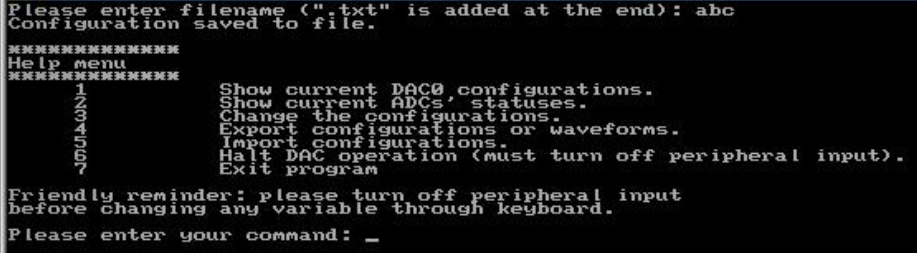
Enter command **“4”** in the help menu to export current configuration, followed by defining the file name of the configuration, as illustrated in Figure 12. The filename has a maximum of 30 characters, including the null terminator. If the filename is NULL, an error message will be printed.

Figure 12: Export Configuration into file.txt

|  |  |
| --- | --- |
| **Keyboard Input:** abc | **Output:** The current configuration of the waveform is saved into file “abc.txt”. |

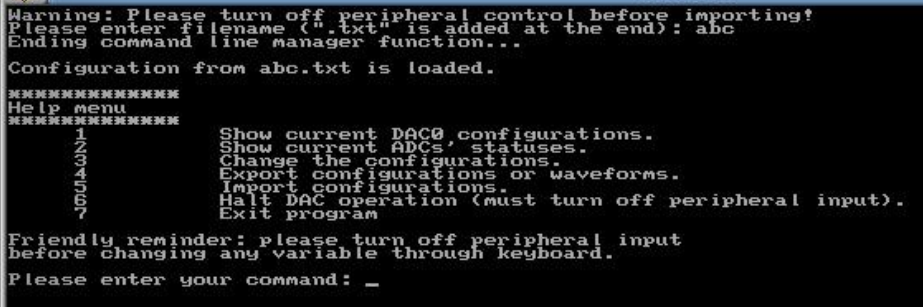
To import the previously saved configuration, the user enters the command **“5”** in the help menu and enters the filename, without “.txt”, as demonstrated in Figure 13.

Figure 13: Importing saved waveform configuration

|  |  |
| --- | --- |
| **Keyboard input:** abc | **Output:** Loads configurations of file “abc.txt” and projects waveform onto oscilloscope. |

## 4.5 Termination using CTRL-C

Figure 14: Termination of program with CTRL-C

In Figure 14, it depicts the response of the program to the user when CTRL-C is entered. It allows the user to return to the program if CTRL-C was pressed unintentionally. If the user wishes to terminate the program through CTRL-C, he/she can enter **“Y”** or **“y”** to exit the program when prompted.

## 4.6 Output waveforms on Oscilloscope

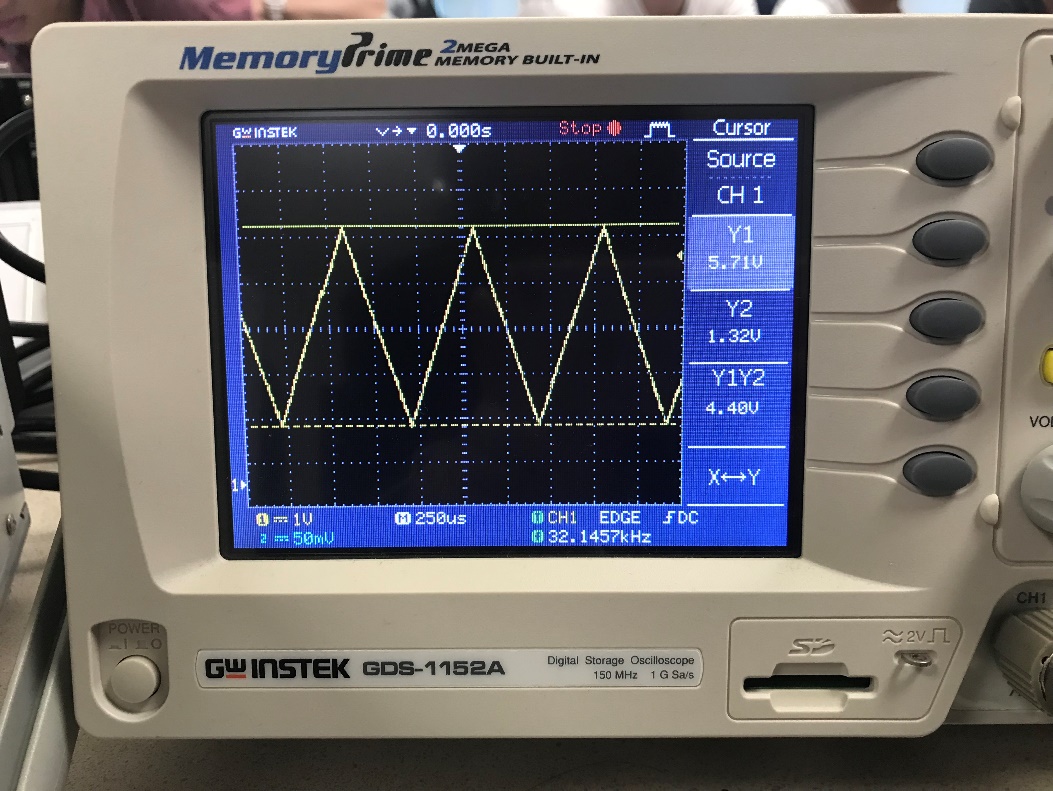
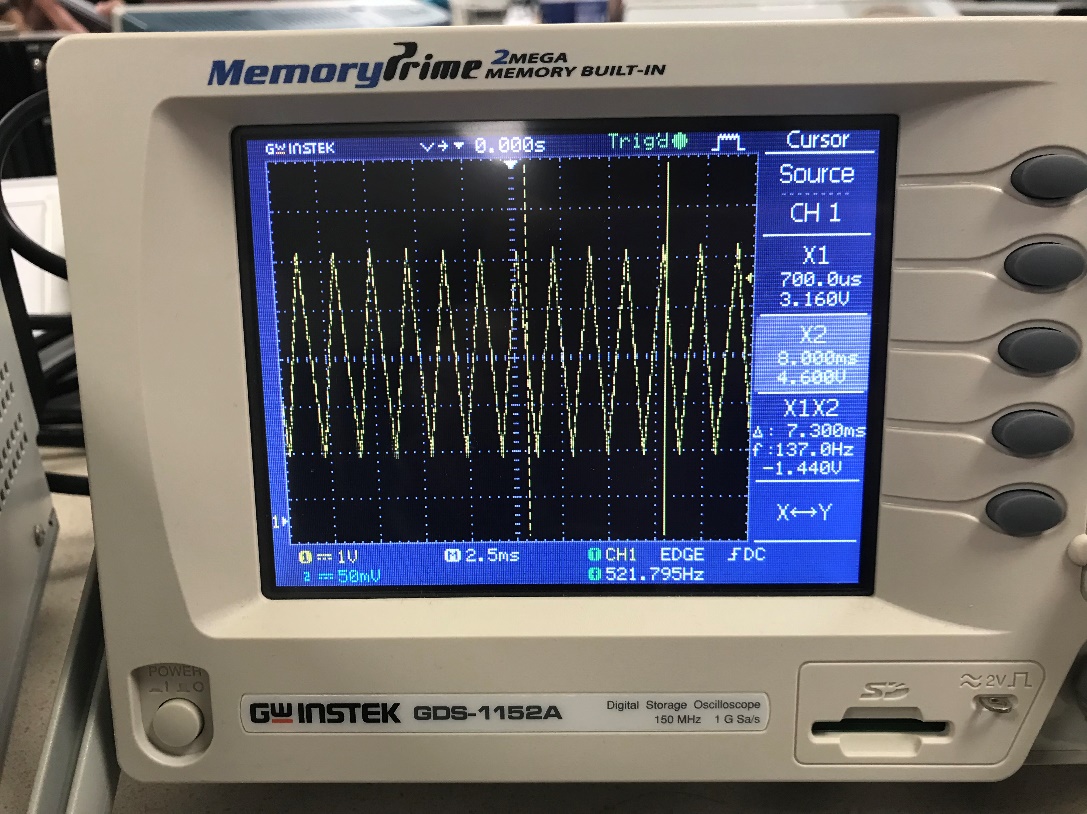
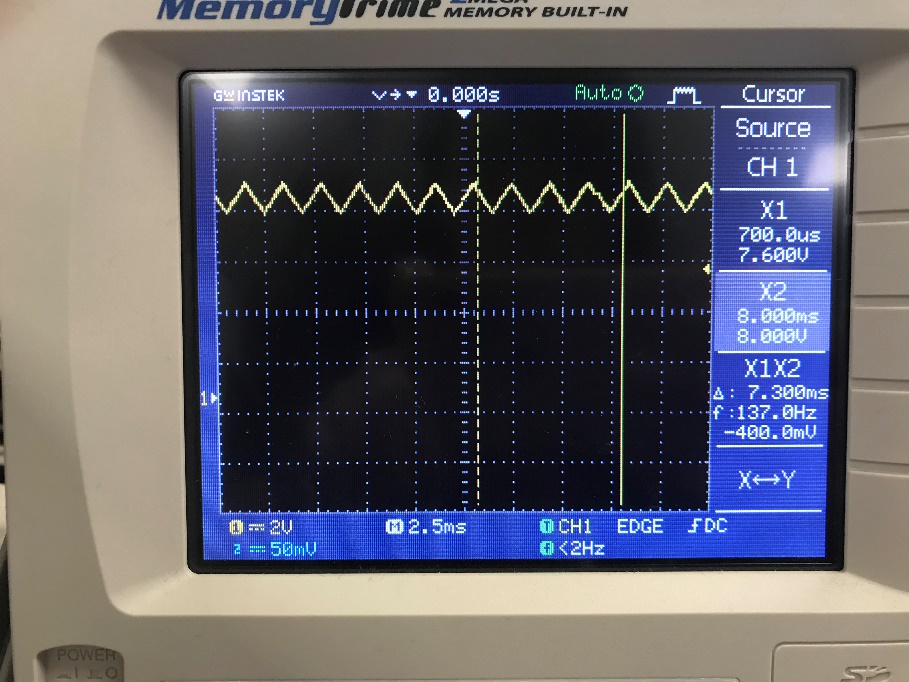
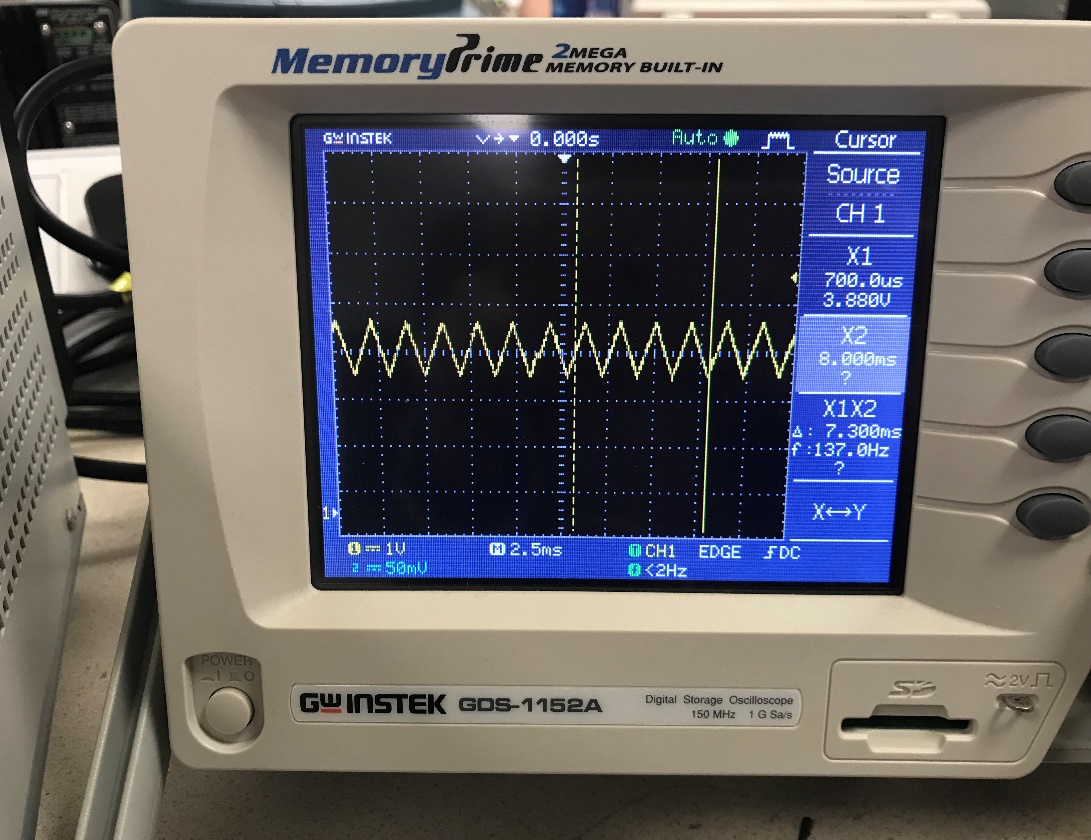
In this section, picture illustrations will show the different actual outputs on the oscilloscope when parameters have been changed.

Figure 18: Changed Frequency, Amplitude & Mean of Original

Figure 17: Changed Frequency & Amplitude of Original

Figure 16: Changed Frequency of Original

Figure 15: Original Triangular Waveform (example)

Figures 15-18 depicts the variations of a triangular wave, which is used as an example.

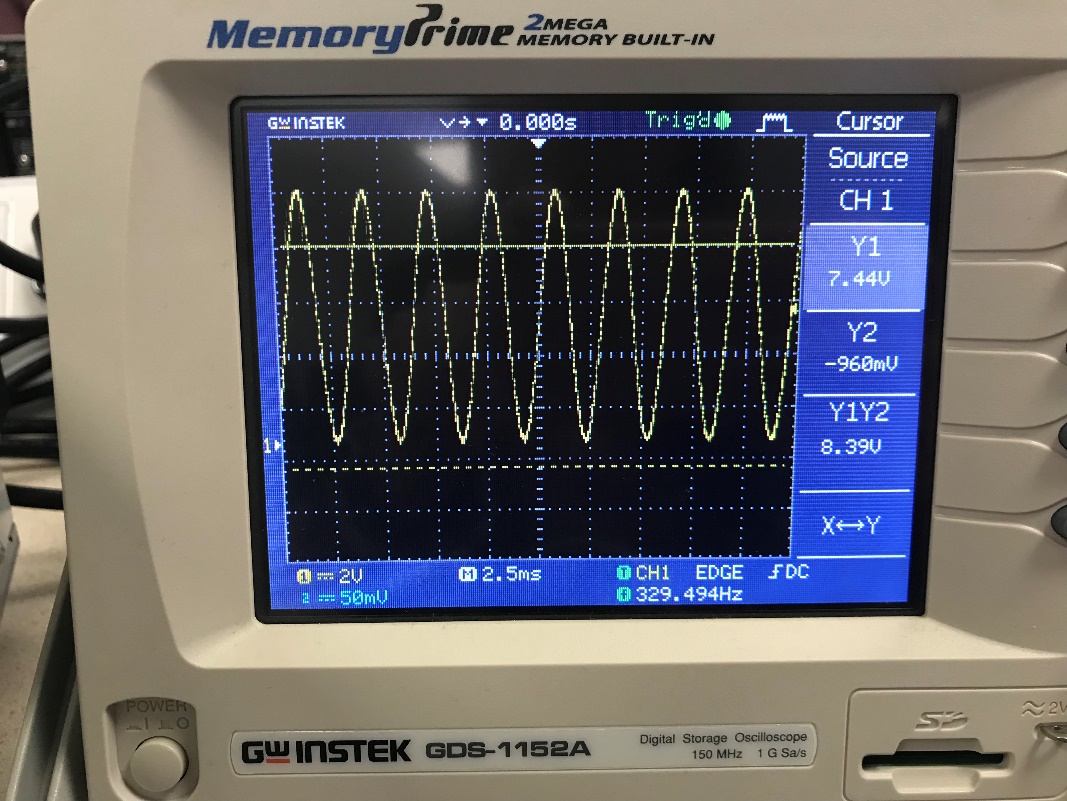
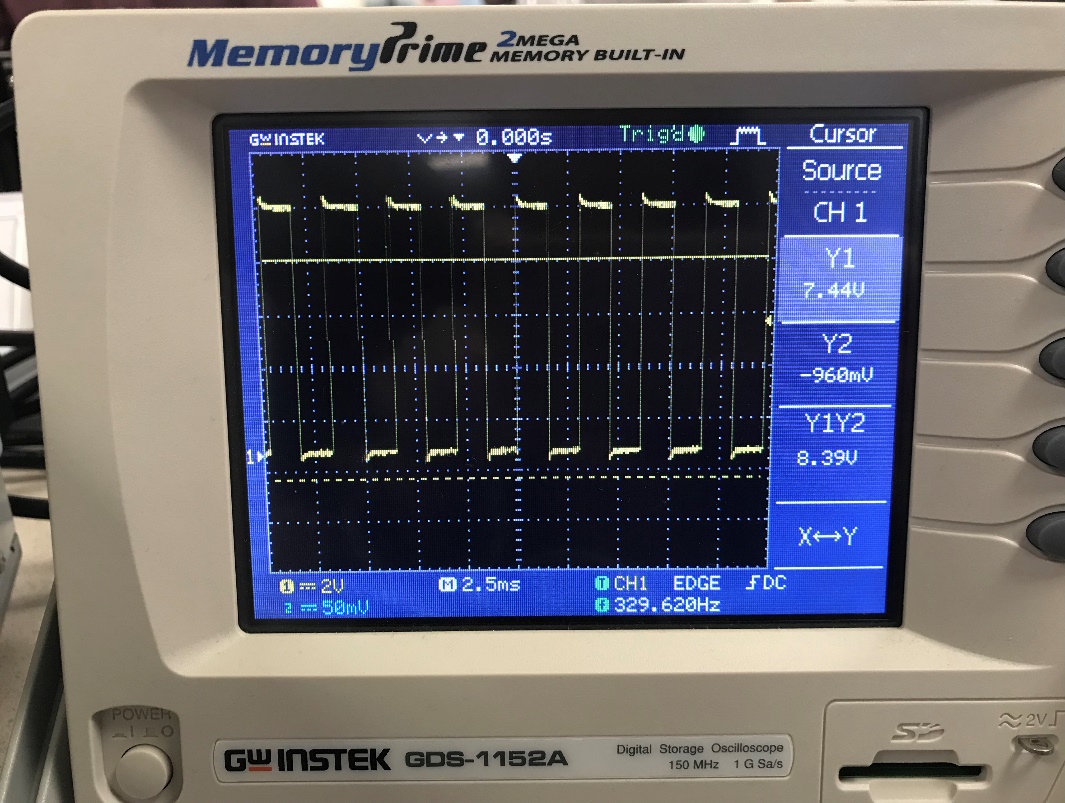
Figures 19-20 illustrates two different waveforms shown on the oscilloscope; the sine wave and square wave, respectively.

Figure 20: Square Wave Output

Figure 19: Sine Wave Output